|  |
| --- |
|  |
|  |
| Performance Test Report |
| Project Name - test 2 |
| Version 1.0 |
| **Date : 9/29/2012** |

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Change Description |
| 9/29/2012 | 1.0 | Amigo Loader | First Generated Report |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Table of Contents |

* **1. Executive Summary.......................................................................................................................1**

1.1 Purpose...........................................................................................................................1

1.2 Objective.........................................................................................................................1

1.3 Project Overview.............................................................................................................1

* **2. Test Environment .........................................................................................................................2**

2.1 Target Servers.......................................................................................................................6

2.2 Deployment Diagram............................................................................................................7

2.3 Load Generators....................................................................................................................7

* **3. Test Results ...................................................................................................................................8**

3.1 Test Conditions and key figures ......................................................................................8

3.2 Load Distribution Profile .................................................................................................8

3.3 Transaction Summary .....................................................................................................8

3.4 Average Response Times .................................................................................................8

3.5 Response Times Vs Time ..........................................................................................8

* **4. Appendix......................................................................................................................................12**

4.1 Appendix 1: ..........................................................................................................................12

4.2 Appendix 2: ..........................................................................................................................13

4.2 Appendix 3: ..........................................................................................................................13

1. Executive Summary

1.1 Purpose

The purpose of this document is to provide a detailed report on the performance test carried out on the test 2 Application. This document provides a summarized view for quick outlookand detailed results including some analysis for the developers to fix any issues. At the end of the report there are Appendices for the terminology used in this document anddeep details from technical aspects.

1.2 Objective

The objectives of this test were to understand the performance and scaling properties of test 2 Application under desired load, on a certain hardware, OS and configurations.This performance testing provided answers to several key questions:

* 1. Can the test 2 application sustain given concurrent user's activities at peak hours?
* 2. What is the projected end-user response time associated with application under load?
* 3. Any bottle neck or performance issues occurring during load test which leads to hinder full utilization of the systems' capacity?

1.3 Project Overview

The applications for the project test 2 is designed for users of  ~~{CUSTOMER NAME} of {DOMAIN}~~ for the purpose of uploading data and manage the complete automation of the business process. Key functionalities of the application that differentiate it are:

* 1. It provides a highly appealing and easy user interface for the  ~~{CUSTOMER NAME} of {DOMAIN}~~  users.
* 2. It has high efficiency of handling automated work flows of ZZZ operations?
* 3. ~~Please add a few functionality of your app.~~

2. Test Environment

2.1 Target Servers

The target servers where the application is deployed along with databases and other servers are as below:

* a) Two App servers with 2.66 GHz 4 CPU and 16 GB RAM each
* b) Two Web servers with 2.66 GHz 8 CPU and 32 GB RAM each
* c) One DB server with 2.66 GHz 8 CPU and 32 GB RAM
* c) One LDAP servers with 2.66 GHz 8 CPU and 32 GB RAM

2.2 Deployment Diagram

The target servers where the application is deployed along with databases and other servers have following deployment architecture:

|  |
| --- |
|  |

2.2 Load Generators

The load generation framework consists of following servers

* a) One controller on a desktop machine having 4 GB RAM and Two CPU of 2.9 GHz
* b) Two load generators on a server machines having 4 GB RAM and Two CPU of 2.33 GHz

3. Test Results

3.1 Test Results Summary

|  |  |
| --- | --- |
| Test Target | To achieve 10 concurrent users load. |
| Test Duration | The test was executed for 1 hr 10 min. |
| Ramp Up Settings | Ramp up was performed at 0 users in every 0 sec for 0 min to achieve max load. |
| Ramp Down Settings | Ramp down was performed at 0 users in every 0 sec for 0 min to end the load. |
| Band Width Emulation | Network speed applied during test was at LAN (Use Max available) Kbps. |
| Total Performed Transactions | 273 |
| Total Passed Transactions | 260 |
| Total Failed Transactions | 13 |
| Pass Percentage | 95.24 % |

3.2 Load Distribution Profile

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction Name | No. of Users | Avg Delay Betn Itrn(s) | Avg Think Time(s) |
| google home | 5 | 15 | 22 |
| google search | 5 | 15 | 22 |

3.3 Transaction Summary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Transaction Name | Min(s) | Max(s) | Avg.(s) | 85%(s) | 90%(s) | 95%(s) |
| Page(Txn)393- | 0.1248 | 0.4058 | 0.17228 | 0.18719 | 0.20273 | 0.20273 |
| Page(Txn)395-Google | 0.34304 | 1.12325 | 0.62415 | 0.79591 | 1.02127 | 1.02127 |
| Page(Txn)405- | 0.5087 | 2.76123 | 1.14563 | 1.79401 | 2.10029 | 2.30396 |

3.4 Average Response Times

|  |
| --- |
|  |

3.5 Response Times Vs Time

|  |
| --- |
|  |

3.6 Dynamic contents vs. Static contents

|  |
| --- |
|  |

3.7 All Static Contents

|  |
| --- |
|  |

4. Appendix  
4.1 Appendix - 1: Terminology

|  |  |
| --- | --- |
| Term / Concept | Description |
| *Capacitiy* | The capacity of a system is the total workload it can handle without violatingpredetermined key performance acceptance criteria. |
| Capacity Test | A capacity test complements load testing by determining your serverâ€™s ultimate failure point, whereas load testing monitors results at various levels of load and traffic patterns. You perform capacity testing in conjunction with capacity planning, which you use to plan for future growth, such as an increased user base or increased volume of data. For example, to accommodate future loads, you need to know how many additional resources (such as processor capacity, memory usage, disk capacity, or network bandwidth) are necessary to support future usage levels. Capacity testing helps you to identify a scaling strategy in order to determine whether you should scale up or scale out. |
| *Component Test* | A component test is any performance test that targets an architectural component of the application. Commonly tested components include servers,databases, networks, firewalls, and storage devices. |
| Endurance Test | An endurance test is a type of performance test focused ondetermining or validating performance characteristics of the product under test when subjected to workload models and load volumes anticipated during productionoperations over an extended period of time. Endurance testing is a subset of load testing. |
| *Investigation* | Investigation is anactivity based on collecting information related to the speed, scalability, and/or stability characteristics of the product under test that may have value indetermining or improving product quality. Investigation is frequently employed to prove or disprove hypotheses regarding the root cause of one or more observedperformance issues. |
| *Latency* | Latency is a measure of responsiveness that represents the time it takes to complete the executionof a request. Latency may also represent the sum of several latencies or subtasks. |
| *Metrics* | Metrics are measurements obtainedby running performance tests as expressed on a commonly understood scale. Some metrics commonly obtained through performance tests include processor utilizationover time and memory usage by load. |
| *Performance* | Performance refers to information regarding your applicationâ€™s responsetimes, throughput, and resource utilization levels. |
| *Performance Test* | A performance test is a technical investigation done todetermine or validate the speed, scalability, and/or stability characteristics of the product under test. Performance testing is the superset containing all othersubcategories of performance testing described in this chapter. |
| *Performance Budgets or Allocations* | Performance budgets (orallocations) are constraints placed on developers regarding allowable resource consumption for their component. |
| *Performance Goals* | Performance goals are the criteria that your team wants to meet before product release, although these criteria may be negotiable under certaincircumstances. For example, if a response time goal of three seconds is set for a particular transaction but the actual response time is 3.3 seconds, it islikelythat the stakeholders will choose to release the application and defer performance tuning of that transaction for a future release. |
| *Performance Objectives* | Performance objectives are usually specified in terms of response times, throughput (transactions persecond), and resource-utilization levels and typically focus on metrics that can be directly related to user satisfaction. |
| *PerformanceRequirements* | Performance requirements are those criteria that are absolutely non-negotiable due to contractual obligations, service levelagreements (SLAs), or fixed business needs. Any performance criterion that will not unquestionably lead to a decision to delay a release until the criterion passesis not absolutely required ? and therefore, not a requirement. |
| *Performance Targets* | Performance targets are the desiredvalues for the metrics identified for your project under a particular set of conditions, usually specified in terms of response time, throughput, andresource-utilization levels. Resource-utilization levels include the amount of processor capacity, memory, disk I/O, and network I/O that your application consumes.Performance targets typically equate to project goals. |
| *Performance Testing Objectives* | Performance testing objectives refer todata collected through the performance-testing process that is anticipated to have value in determining or improving product quality. However, these objectives arenotnecessarily quantitative or directly related to a performance requirement, goal, or stated quality of service (QoS) specification. |
| *PerformanceThresholds* | Performance thresholds are the maximum acceptable values for the metrics identified for yourproject, usually specified in terms of response time, throughput (transactions per second), and resource-utilization levels. Resource-utilization levels include theamount of processor capacity, memory, disk I/O, and network I/O that your application consumes. Performance thresholds typically equate to requirements. |
| *Resource Utilization* | Resource utilization is the cost of the project in terms of system resources. The primary resourcesare processor, memory, disk I/O, and network I/O. |
| *Response Time* | Response time is a measure of how responsive an applicationorsubsystem is to a client request. |
| *Saturation* | Saturation refers to the point at which a resource has reached fullutilization. |
| *Scalability* | Scalability refers to an applicationâ€™s ability to handle additional workload, without adversely affectingperformance, by adding resources such as processor, memory, and storage capacity. |
| *Scenarios* | In the context of performancetesting, a scenario is a sequence of steps in your application. A scenario can represent a use case or a business function such as searching a productcatalog, adding an item to a shopping cart, or placing an order. |
| *Smoke Test* | A smoke test is the initial run of a performancetest to see if your application can perform its operations under a normal load. |
| *Spike Tes*t | A spike test is a type ofperformance test focused on determining or validating performance characteristics of the product under test when subjected to workload models and load volumes thatrepeatedly increase beyond anticipated production operations for short periods of time. Spike testing is a subset of stress testing. |
| *Stability* | In the context of performance testing, stability refers to the overall reliability, robustness, functional and dataintegrity, availability, and/or consistency of responsiveness for your system under a variety conditions. |
| *Stress Test* | A stresstestis a type of performance test designed to evaluate an applicationâ€™s behavior when it is pushed beyond normal or peak load conditions. The goal of stresstesting is to reveal application bugs that surface only under high load conditions. These bugs can include such things as synchronization issues, race conditions,and memory leaks. Stress testing enables you to identify your applicationâ€™s weak points, and shows how the application behaves under extreme load conditions. |
| *Throughput* | Throughput is the number of units of work that can be handled per unit of time; for instance, requests persecond, calls per day, hits per second, reports per year, etc. |
| *Unit Test* | In the context of performance testing, a unit test is anytest that targets a module of code where that module is any logical subset of the entire existing code base of the application, with a focus on performancecharacteristics. Commonly tested modules include functions, procedures, routines, objects, methods, and classes. Performance unit tests are frequentlycreatedand conducted by the developer who wrote the module of code being tested. |
| *Utilization* | In the context of performancetesting, utilization is the percentage of time that a resource is busy servicing user requests. The remaining percentage of time is considered idle time. |
| *Validation Test* | A validation test compares the speed, scalability, and/or stability characteristics of the product under testagainst the expectations that have been set or presumed for that product. |
| *Workload* | Workload is the stimulus applied to a system, application, or component to simulate a usage pattern, in regard to concurrency and/or data inputs. The workload includes the total number of users, concurrent active users, data volumes, and transaction volumes, along with the transaction mix. For performance modeling, you associate a workload with an individual scenario. |